Perfect simulation of a class of stochastic hybrid systems with an application to peer to peer systems

Bruno Gaujal, Florence Perronnin, Remi Bertin

Perfect simulation of a class of stochastic hybrid systems with an application to peer to peer systems

Bruno Gaujal* Florence Perronnin† Rémi Bertin‡

October 18, 2013

Abstract

In this paper we present a class of hybrid systems made of deterministic differential equations and random discrete jumps. We then show how to construct a simulation of such a stochastic hybrid system that provides perfect samples of its asymptotic behavior based on the extension to continuous state-space of coupling-from-the-past techniques introduced by Foss and Tweedie (1998) and using suitable envelope trajectories to tackle non-monotonicity.

The applicability of the method is illustrated by showing how this framework can be used to model the Squirrel peer to peer system and by reporting a simulation study based on this approach.

This paper provides both a framework on how to carry simulation based experimental studies of large and complex hybrid systems and its application in the Squirrel model demonstrating how versatile and powerful this approach can be over a typical example.

1 Introduction

Hybrid systems have become popular to model systems whose dynamics is an interplay between a continuous one and a discrete one.

The main difficulty in dealing with them is that continuous systems are often modeled by differential equations while discrete ones are in general seen as automata. The mathematical tools to analyze the two parts (calculus versus algebra) as well as computer issues (numerical precision versus complexity) from both worlds are hardly compatible. However several recent breakthroughs

*INRIA and LIG (CNRS, INPG, INRIA, UJF) 51 Av. J. Kunztmann, Montbonnot, France. Email: Bruno.Gaujal@imag.fr

†UJF and LIG (CNRS, INPG, INRIA, UJF) 51 Av. J. Kunztmann, Montbonnot, France. Email: Florence.Perronnin@imag.fr

‡Supelec, Metz. This work was done while this author was visiting the Laboratory ID-IMAG under an INRIA internship

such as [Alur et al., 2003, Tomlin et al., 2003, Girard, 2005] have demonstrated the usefulness and the interest from both theoretical and practical points of view, of mixing the continuous and the discrete paradigms.

Historically, the most widely used hybrid systems are *switched* systems. These systems can be classified into state-dependent and time-dependent systems [Liberzon, 1973]. Consider a family $\mathcal{P}$ of differential equations $\{\dot{x} = f_p(x),\, p \in \mathcal{P}\}$. A *time-dependent* switched system is composed of a piecewise continuous process x and a piecewise-constant function $\sigma : \mathbb{R}+ \mapsto \mathcal{P}$ verifying $\dot{x} = f_{\sigma(t)}(x)$. In a *state-dependent* switching system, the index p of function f_p depends on x itself and not on a time variable. This results in a process hitting boundaries (switching surfaces) across which the differential system changes.

The main focus in the literature is on *stability analysis* (i.e., find conditions such that the system remains in a compact domain) and *control synthesis* (find control policies such that the system remains stable) [Liberzon, 1973].

More recently, other classes of hybrid systems have been studied such as hybrid stochastic processes where the differential equations and/or the discrete dynamics are governed by stochastic inputs. Examples of such systems are piecewise-deterministic Markov processes (PDMP) [Davis, 1984], where the continuous part is deterministic and the discrete part is Markovian, or switched diffusion processes where the continuous part involves a Brownian term [Ghosh et al., 1997]. In such systems, the vast majority of research is on the study of the long-term behavior of the system (recurrence, ergodicity) and, for controlled systems, on the existence of simple optimal control policies.

In this paper, we will rather focus on *computational issues*, i.e., find efficient algorithms to compute numerically the asymptotic behavior of hybrid systems which are non-homogeneous PDMPs with continuous state-spaces.

The analysis of such systems is often difficult on a mathematical as well as on a numerical point of view and such large hybrid systems are often considered computationally untractable (even in the simple case where the PDMP is homogeneous and the state-space is discrete). Simulation approaches are efficient alternative methods to estimate the behavior of such systems by providing samples distributed according to its asymptotic distribution, even when it is impossible to compute this distribution numerically. However, simulation has several drawbacks. First, simulations do not make any sense unless the system has ergodicity properties, which are sometimes difficult to assert. Second, even under the right ergodicity conditions, classical simulation techniques only provide approximations of the asymptotic behavior. The longer the simulation the more accurate the result, but it is usually hard or impossible to be more precise than this general statement. Recently, Propp and Wilson used backward coupling techniques ([Propp and Wilson, 1996]) to design a simulation algorithm to get *perfect sampling* (*i.e.* whose distribution is not an approximation

but the exact asymptotic distribution) of discrete time finite Markov chains. Their technique has been extended to Markov chains over continuous state spaces under uniform ergodicity conditions in [Foss and Tweedie, 1998] and we show how this is suitable for hybrid systems by using the embedded chain at jump instants.

More precisely, in this paper, we show that backward coupling technique can be used to simulate perfectly an *embedded Markov chain* in a stochastic hybrid system. We show that when a partial order exists on the state space such that the evolution equations verify some monotonicity properties, or when envelopes of all trajectories couple in finite time, then backward coupling can be done by simulating only a small number of trajectories, starting with the maximal and the minimal states.

These simulations provide, in finite time, trajectories of a hybrid system which follow its asymptotic behavior. Our simulation algorithm is presented under a general framework of a hybrid system made of deterministic ordinary differential equations coupled with a Markovian discrete process. To illustrate the approach and to give an insight on the applicability of the method, an example of a hybrid model of the squirrel peer to peer (P2P) system [Iyer et al., 2002] is developed in full details throughout the paper.

This real-world system is actually fully discrete but involves several time and space scales (see Section 3). In that case, hybrid systems are very useful: one can use *fluid limits* for the parts of the system with fastest and largest scales to make them continuous, and keep the slow part discrete. These models have been introduced in various domains under the form of fluid queues [Dai, 1995], continuous Petri nets [David and Alla, 2004], or timed automata [Asarin et al., 1995]. In this paper, we will consider one such example, namely peer to peer systems, where two types of dynamics are superimposed. The slow dynamics concerns the customers, who join and leave the system. The fast dynamics concerns the files and their transfers between the customers. While hybrid systems often provide compact and elegant models for complex systems

The goal of this paper is two-fold. First it provides a general framework on how to carry experimental studies based on perfect simulations, of large and complex hybrid systems. Second, it shows how fast, versatile, practical and powerful this approach can be over a typical example, namely the Squirrel peer to peer system, which exhibits many features which make hybrid systems difficult to study: a large state space, highly non-linear dynamics and an intricate interplay between its discrete and continuous parts.

The paper is organized as follows. In Section 2, we present the general hybrid system for which a perfect simulation algorithm will be given and construct the corresponding embedded Markov chain. Section 3 describes the Squirrel model in details and provides the corresponding hybrid model fitting the general framework. Section 4 shows how hybrid systems are amenable

to perfect simulations under several assumptions: the state-space admits a partial order and the Markov chain is ergodic in a strong sense. We also provide a general algorithm even when the monotonicity of the embedded Markov chains is not verified. This problem is by-passed by simulating lower and upper envelopes instead, which have the wanted monotonicity property and couple in finite time. All the concepts in this section are illustrated with the Squirrel instance. Finally Section 5 reports a complete study of the Squirrel model based on the perfect simulation tool implementing the envelope algorithm and Section 6 provides several improvements in terms of simulation time and fast approximations.

2 Stochastic hybrid systems

In this paper we consider a dynamic system $\mathcal{S}$ whose state is a couple $(N(t), x(t))$ where $N(t)$ is the discrete component of $\mathcal{S}$ and can only take discrete values while x is the continuous part of $\mathcal{S}$. The variables $(N(t), x(t))$ will be cadlag functions in the following.

Classical issues from control (observability, controllability) are not essential (or become trivial) here by compactness of the state space. Most stochastic switched system issue (ergodicty, stationnarity) will be assumed to be verified. Here we tackle computational problems both on an algorithmic point of view and a practical ones: get numerical guarantees for the asymptotic behavior for huge systems.

The discrete process $N(t)$ evolves in a finite, discrete state space $M \subset \mathbb{N}$. $N(t)$ only changes values at random times $(T_n)_{n\geq 0}$, that forms a point process whose rate $\nu(t)$ depends on both the discrete and continuous components before the jump instant: $\nu(t) = \psi(N(t^-), x(t^-))$. The destination state of $N(t)$ after the jump is a random variable whose distribution $Q(i,j)_{i,j\in M^2}$ only depends on $N(t) = i$. We further assume that the function ψ is bounded so that the jump rate is bounded, which ensures the absence of Zeno behaviors.

The continuous process $x(t)$ evolves in a compact $D \subset \mathbb{R}^n$ according to a deterministic differential equation that depends on $N(t)$, which is constant between consecutive jumps:

$$\frac{dx}{dt} = f(N(T_n), x, t). \tag{1}$$

At jump times T_n, $x(t)$ has discrete, stochastic jumps δ_n whose distribution $\Delta_{N,x}$ depends on N and x just before the jump.

Such dynamical equations couple the discrete and continuous part in both directions and therefore do not belong to the class of simple switched systems. In our framework, the switching is both time and space dependent, in a stochastic sense.

There are classes of stochastic hybrid systems that cannot be described in this framework: those with non Poisson jumps and/or with continuous stochastic dynamics, such as Brownian

motions.

2.1 Embedded Markov chains

In the following, we will restrict our study to cases with an integer discrete part $N(t) \in \mathbb{N}$, a real continuous part $x(t)$ which is mono-dimensional ($x(t) \in \mathbb{R}$) and a differential equation $\frac{dx}{dt} = f(N(T_n), x, t)$ that admits a unique solution for all possible initial states (N_0, x_0), denoted $F(N_0, x_0, t)$.

We denote by $\mathcal{D} \subset \mathbb{N} \times \mathbb{R}$ the set of all reachable states for the couple (N, x).

Lemma 1. *Under the foregoing assumptions, the embedded process $S_n = (N(T_n), x(T_n))$ is a homogeneous discrete-time Markov chain over the continuous domain, $\mathcal{D} \subset \mathbb{N} \times \mathbb{R}$.*

Proof. It should be clear from the definition of N and x that for all measurable set A in $\mathcal{D}$,

$$\begin{aligned}&\mathbb{P}((N(T_{n+1}), x(T_{n+1})) \in A | (N(T_n), x(T_n)) = (a_n, b_n), \ldots, (N(T_0), x(T_0)) = (a_0, b_0)) \\ &= \mathbb{P}((N(T_{n+1}), x(T_{n+1})) \in A | (N(T_n), x(T_n)) = (a_n, b_n)).\end{aligned}$$

The state of the process at time T_{n+1} only depends on the state at time T_n and the n-th inter-arrival of the jump process $T_n - T_{n-1}$, whose value only depends on the state at time T_{n-1}. This means that $(N(T_n), x(T_n))$ is a Markov chain over the domain of all reachable states, $\mathcal{D} \subset \mathbb{N} \times \mathbb{R}$. □

3 Overview and model of Squirrel

Throughout the paper, we will use the example of a hybrid system to illustrate and to apply the general framework. The example models a peer to peer caching system. We believe that the example provides a good illustration of the applicability of the general framework, but also presents a significant interest *per se*, providing interesting insights on the evolution of a peer to peer system. Indeed, in Section 5 we use our approach to compute detailed performance metrics of complex P2P systems, such as the probability that files are present in the system.

In this section we briefly introduce the running example, namely Squirrel: a P2P cooperative Web cache. Then we show how it can be modeled as a hybrid system.

Squirrel system description

Web caching consists in keeping copies of Web objects *locally* or *closer* to the end-users than the origin servers, so as to limit bandwidth consumption and download time. In the context of Web caching, the "servers" may be a local proxy or even a local cache on each client's machine. The Squirrel system [Iyer et al., 2002] was designed to enable users to share their local Web caches to make a distributed P2P caching system.

We now briefly describe how Squirrel works. The interested reader can refer to [Iyer et al., 2002] for a complete description of the protocol. In the following, a *node* will denote a user's machine. Squirrel is based on the Pastry routing protocol [Rowstron and Druschel, 2001]. Nodes emit *requests* for Web objects. The local cache is then checked first. If the object is not found ("cache miss") the client tries to locate the file on another node of the Squirrel system as follows. The URL of the object is mapped to a number called "object-Id" using a hash function. The request is then forwarded using Pastry to the node whose node-Id is numerically closest to the object-Id. This latter node is called "home node" for this object. The home node acts as a classical cache for this object : it checks its own local cache for the object. In case of a cache hit the object is sent to the requesting node. In case of a cache miss, the home node downloads the object from the origin Web server, stores a copy locally and sends it to the requesting node.

Finally, nodes can join and leave the system at anytime, for instance due to peer disconnection or to software crash. When a node A leaves the system, the objects that were cached in A are lost for the whole system. When a node B joins the system, it becomes *de facto* a home node for a number of objects. We assume that B does not bring any exogenous content to the global system when joining in [Clévenot-Perronnin, 2005].

Modeling Squirrel with a hybrid system

We now introduce a fluid-discrete model for Squirrel. This model was introduced and experimentally validated in [Clévenot-Perronnin, 2005]. Let us now describe the model and show that it belongs to the class of hybrid systems defined in Section 2.

We assume that nodes join and leave independently of each other. This process is the discrete component of the model. Let $N(t)$ denote the number of connected nodes. We assume the system is closed, i.e., there may exist only N_{max} nodes. Each node leaves the system with a constant rate μ. Each node joins the system with a constant rate λ. Note that this process is a classical continuous time Markov chain whose generator is displayed in Figure 1. This is the well-known Ehrenfest urn model which is studied in [Kelly, 1979] for example.

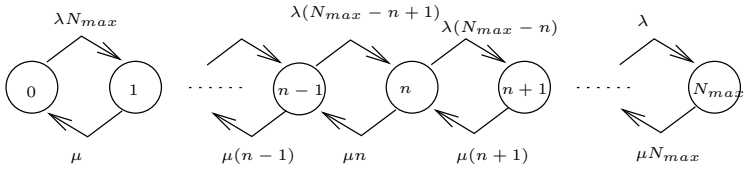

Figure 1: The infinitesimal generator for the Markov process N

We now model the files dynamics in the Squirrel system.

In the following we assume that the request rate σ is constant and identical for each node.

The total amount of cached documents is modeled with a fluid system.

The justification for this continuous model is that files may be partially transferred and also that these transfers occur on a much faster time scale than node events. More details on this fluid assumption can be found in [Clévenot-Perronnin, 2005]. Specifically, let $x(t)$ denote the total amount of fluid, i.e. the total number of files in the Squirrel system. This process is the continuous component of our hybrid system. Again, we assume a closed set of objects, i.e. $x(t)$ is bounded by the maximum number of existing files C.

We now describe the dynamics of $x(t)$ and show how they can be modeled by Equations (6)-(1). Let us first determine the differential equation which describes the evolution of fluid between two consecutive jumps of $N(t)$. The amount of fluid increases whenever a client brings a new file in the system. That is, upon a cache miss, the home node downloads the requested file and stores it for future requests. This increases $x(t)$. This process is proportional to the total request rate $\sigma \times N(t)$ and to the miss probability. Using the well-known Zipf-like popularity distribution of Web objects [Breslau et al., 1999], we show in Appendix A.1 that the miss probability is of the form $1-\left(\frac{x(t)}{C}\right)^{\alpha}$ with $1 \geq \alpha > 0$. Finally the fluid decreases when copies become stale or obsolete, typically after an average time-to-live $1/\theta$. As a result, the function f in differential equation (1) becomes homogeneous in time and can be expressed as::

$$\frac{dx}{dt} = f(N(T_n), x(t)) = \sigma N(T_n)\left(1 - \left(\frac{x(t)}{C}\right)^{\alpha}\right) - \theta x(t). \tag{2}$$

When $\alpha = 1$ this ODE admits a closed-form solution and the stationary hit probability of the Squirrel system may be analytically expressed in closed form [Clévenot and Nain, 2004]. However, if $\alpha \neq 1$ this equation admits a unique solution (see Appendix A.3) with no known closed-form and a numerical resolution is needed.

We now turn to the evolution of the fluid at jump instants T_n. When a node leaves the system it takes away all the documents it was responsible for. With the load balancing property of Pastry, the number of lost documents is a proportional fraction of the total fluid. Therefore if the event at T_n is a departure then

$$x(T_n) = \frac{N(T_{n-1}) - 1}{N(T_{n-1})}\, x(T_n^-). \tag{3}$$

When a node joins the system, it does not bring exogenous files with him upon its arrival. The increase of the number of files will come from the future downloads of the newcomer:

$$x(T_n) = x(T_n^-). \tag{4}$$

Figure 2 shows a typical trajectory of the evolution of both N and x over time. By looking closely to the evolution of $x(t)$, one may notice down jumps corresponding to departures of customers and singularities (instants where x remains continuous but is not differentiable), corresponding to customer arrivals.

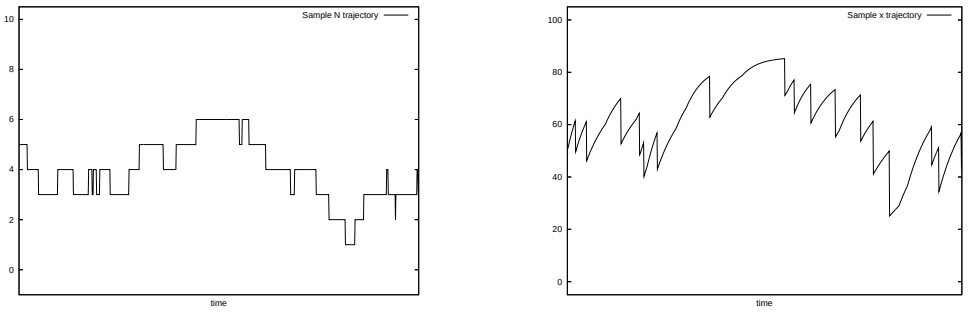

Figure 2: A trajectory of N and x respectively for the Squirrel model, with $C = 100, N_{max} = 10, \lambda = \mu = 1, \theta = \sigma = 1, \alpha = 1$.

The Squirrel example does not exhibit all the complexity of our general framework. Indeed, the discrete part is autonomous (it does not depend on the continuous state). Also, both the discrete and continuous parts are of dimension 1 ($\mathbb{Z}$ and $\mathbb{R}$ respectively). A more complex application where the discrete part depends on the continuous state is presented in [Gaujal and Perronnin, 2007]. The framework applies similarly in that case.

A model with a high dimension but with very simple continuous dynamics is studied in [Vincent, 2005]. Despite the high dimensionality, two envelope trajectories are enough in that case to provide perfect sampling of the system.

4 Perfect simulation of Markov chains over continuous state spaces

4.1 Sampling the steady-state distribution of a Markov chain

As mentioned in the introduction, the goal of this paper is to get guaranteed estimates of the asymptotic behavior of the hybrid system $\mathcal{S}$. In order to do so, we will provide a *perfect simulation algorithm* of the Markov chain $S_n = (N(T_n), x(T_n))$.

Let us first recall the main ingredients for perfect sampling of Markov chains. Let $P^n(s, A)$ denote the transition law of n steps of the chain S_n (This is the probability $\mathbb{P}(S_n \in A | S_0 = s)$). When it exists, its stationary measure satisfies $\Pi(A) = \int_{\mathcal{D}} P^1(s, A)\Pi(ds)$, for all measurable set A in $\mathcal{D}$.

Of course, if Π can be computed explicitly, there are many ways to draw samples from it. However, in most cases, analytical or even numerical computations of Π are impossible to obtain, either because the domain $\mathcal{D}$ is huge (in finite cases) or because the structure of the kernel $P^1(s, A)$ is too complex.

Without analytical or numerical knowledge of Π the most popular method for sampling from Π is simulation. The classical Monte-Carlo simulation consists in choosing an arbitrary initial value $S_0 = s_0$ in $\mathcal{D}$ and to use the constructive equations given below in Equations (5),(6) and (7), to generate $S_1, \ldots, S_n$ using a random number generator. This technique works asymptotically because the sequence of samples converges in law, in the sup-norm, to the stationary distribution:

$$\lim_{n\to\infty} \sup_A |P^n(s_0, A) - \Pi(A)| = 0.$$

However, for a given finite n, the gap with the exact distribution depends on the convergence rate to the stationary distribution which is unknown in general. Therefore, it is difficult to estimate the bias between $P^n(s_0, A)$ and $\Pi(A)$ in general.

Here, we will show how to compute samples in finite time whose distribution is *exactly* Π (hence the name perfect), using a backward coupling technique. This technique was proposed for the first time in [Propp and Wilson, 1996] for Markov chains over finite state spaces. The main idea is to run several simulations in parallel *starting in the past* from all possible initial states and look at what happens at time 0. If all the trajectories coincide at time 0, then the simulation stops and outputs the common value of all the trajectories at time 0, which happens to be a perfect sample of the chain.

This idea was extended to Markov chains over continuous state spaces in [Foss and Tweedie, 1998] by proving the existence of a finite vertical backward coupling time.

In this paper we use their theorem to construct an exact sampling algorithm for hybrid models and we show how to deal with continuous systems that are not strictly monotonic.

We first propose a constructive definition of the Markov chain that is needed for algorithmic purposes.

4.2 Constructive Markov Chain

We now introduce functions φ and h to describe the Markov chain in a constructive way. This can be done with no loss of generality.

The idea used in the following construction is to simulate a variable-rate Poisson process with rate $\lambda(t) \leq \Lambda$ by generating a Poisson process $0 \leq P_1 \leq P_2 \leq \cdots \leq P_n \leq \cdots$ with rate Λ and then by removing each point P_i with probability $1 - \frac{\lambda(P)}{\Lambda}$.

Here, the function ψ is bounded (by, say, Λ), it is possible to generate the jump times T_{n+1} as a sub-sequence of a Poisson point process $\Upsilon_1, \Upsilon_2, \ldots$ with rate Λ in the same way, that is by setting $T_{n+1} = \Upsilon_{\delta(n+1)}$ where $\delta(n+1) = i$ with probability

$$\frac{\psi(N(T_n), F(N(T_n), x(T_n), \Upsilon_i - T_n))}{\Lambda} \prod_{j=\delta(n)+1}^{i-1} 1 - \frac{\psi(N(T_n), F(N(T_n), x(T_n), \Upsilon_j - T_n))}{\Lambda}$$

In other words, $\Upsilon_{\delta(n)+1}$ corresponds to the next real jump T_{n+1} with probability

$$\frac{\psi(N(T_n), F(N(T_n), x(T_n), \Upsilon_{\delta(n)+1} - T_n))}{\Lambda},$$

and otherwise, the next point $\Upsilon_{\delta(n)+2}$ is considered, and so forth. Note that $\delta(n+1)$ is finite as long as T_{n+1} is well defined.

The process $N(t)$ is driven by those potential *jump instants*, which form a point process $\Upsilon_0 = 0$ (time origin), $\Upsilon_1, \ldots \Upsilon_n, \ldots$. At time Υ_n, one can write as for any Markov chains,

$$N(\Upsilon_n) = \varphi(N(\Upsilon_{n-1}), x(\Upsilon_n^-), \xi_n), \tag{5}$$

where $\{\xi_n\}_{n\in\mathbb{N}}$ is a random process of *innovations* (a random variable uniformly distributed over $[0,1]$) and φ is a deterministic function that describes the dynamics of N at jump instants.

As for the continuous part, $x(t)$ is governed by a process at jump instants (using the same innovation ξ_n),

$$x(\Upsilon_n) = h(N(\Upsilon_{n-1}), x(\Upsilon_n^-), \xi_n), \tag{6}$$

and by a deterministic differential equation between jump instants. In $[\Upsilon_n, \Upsilon_{n+1})$,

$$\frac{dx}{dt} = f(N(\Upsilon_n), x, t). \tag{7}$$

This defines a global function Φ as $(N(\Upsilon_n), x(\Upsilon_n)) = \Phi(N(\Upsilon_{n-1}), x(\Upsilon_{n-1}), \xi_n, \tau_n)$, with $\tau_n = \Upsilon_n - \Upsilon_{n-1}$.

Instantiating with Squirrel: To deal with the continuous time Markov chain driving N, the uniformization procedure described above is applied using the uniformization constant $\Lambda = (\lambda + \mu)N_{max}$. The transition matrix of this embedded discrete time Markov chain is given in Figure 3, where the rates are normalized: $\lambda' = \lambda/\Lambda$ and $\mu' = \mu/\Lambda$.

Using uniformization, the stochastic process will be considered at times Υ_n which include all real jumps T_n as well as fictitious jumps (null events).

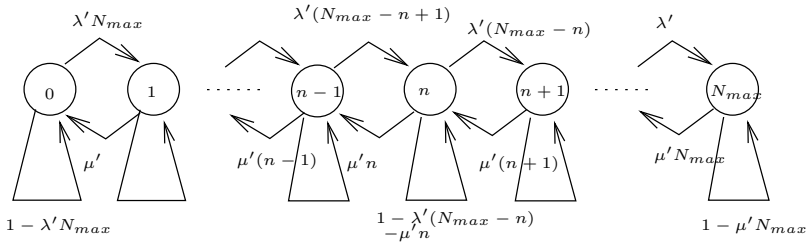

Figure 3: The transition kernel of the uniformized Markov chain

Note that in this example, N does not depend on x so that its evolution can be computed in isolation. We would like to emphasize that while this greatly simplifies the computation for this

particular instance, it is not a necessary condition for our framework, and more general hybrid systems have been analyzed similarly in [Gaujal and Perronnin, 2007].

Also note that N being a one-dimensional random walk, it is an ergodic chain. Its stationary distribution π_E is easy to compute using the closed-form formula:

$$\pi_E(N=k) = \frac{\left(\frac{\lambda}{\mu}\right)^k \binom{N_{max}}{k}}{\left(1+\frac{\lambda}{\mu}\right)^{N_{max}}} \quad \forall\, 0 \leq k \leq N_{max}. \tag{8}$$

However, the asymptotic behavior of the continuous part, x is more difficult to compute for several reasons. First, the differential equation (2) does not have a closed form solution when α is not an integer. Second, the stochastic jumps and the dependence with respect to N make the asymptotic behavior of x hard to grasp, even numerically.

Here is the functional construction of the next state $N(\Upsilon_n), x(\Upsilon_n)$ of the chain using $N(\Upsilon_{n-1})$ and $x(\Upsilon_n^-)$. This corresponds to defining the functions φ and h.

1. Generate ξ uniformly over $[0,1]$.
2. Generate τ exponentially distributed with parameter Λ.
3. $\Upsilon_n = \Upsilon_{n-1} + \tau$.
4. Integrate the differential equation over span τ: $x(\Upsilon_n^-) = F(x(\Upsilon_{n-1}), N(\Upsilon_{n-1}), \tau)$.
5. - If $\xi > 1 - \lambda'(N_{max} - N(\Upsilon_{n-1}))$, then the next event is a *customer arrival:* $N(\Upsilon_n) = N(\Upsilon_{n-1}) + 1$ and $x(\Upsilon_n) = x(\Upsilon_n^-)$.
   - If $\xi < \mu' N(\Upsilon_{n-1})$, then the next event is a *customer departure:* $N(\Upsilon_n) = N(\Upsilon_{n-1}) - 1$ and $x(\Upsilon_n) = x(\Upsilon_n^-) - x(\Upsilon_n^-)/N(\Upsilon_{n-1})$.
   - Otherwise, this is a *null event, i.e.* no customer arrives nor leaves and the system is left unchanged at time Υ_n: $N(\Upsilon_n) = N(\Upsilon_{n-1})$ and $x(\Upsilon_n) = x(\Upsilon_n^-)$.

In most cases, the integral F of the differential equation cannot be computed under closed form. In this case, we use a classical Runge-Kutta numerical integrator [Press et al., 1992] to compute $x(\Upsilon_{n+1}^-)$.

4.3 Backward coupling of Continuous state space Markov chains

Here is the main theorem (for our use) in [Foss and Tweedie, 1998], adapted to our notations.

Theorem 2 (Vertical backward coupling [Foss and Tweedie, 1998])**.** *If the Markov chain is uniformly ergodic, i.e. if there exists a probability measure γ over $\mathcal{D}$, some $m > 1$ and $0 < B \leq 1$*

such that

$$\forall x \in \mathcal{D} \quad P^m(x, \cdot) \geq B\gamma(\cdot),$$

then, the vertical backward coupling time

$$K \stackrel{\text{def}}{=} \min\{n \geq 0 : \Phi(s, \xi_{-n}, \tau_{-n} \ldots, \xi_{-1}, \tau_{-1}) = \Phi(r, \xi_{-n}, \tau_{-n} \ldots, \xi_{-1}, \tau_{-1}), \forall r, s \in \mathcal{D}\},$$

is a well defined random variable. Furthermore, $\forall s \in \mathcal{D}$, $\Phi(s, \xi_{-n}, \tau_{-n} \ldots, \xi_{-1}, \tau_{-1}) \sim \Pi$.

Note that K is defined by using *backward* iterations of function Φ. Instead of starting from one initial state at time 0 and make the chain evolve from that point on, here, the iterations start at time $-K$ using all possible initial states and evolve up to time 0 where they should all coincide (or couple). This is the reason why this technique is also called "coupling from the past".

Note that starting at all initial states and simulating forward until coupling of all trajectories gives a biased result since trajectories may couple at specific states (in the Squirrel instance it is $(0, 0)$), where a forward simulation would stop. Simulating from the past stops the simulation at time 0 if and only if all trajectories coincide, which provides a state with the stationary distribution.

Also note that Theorem 2 does not provide directly an effective algorithm for computing perfect samples $\Phi(s, \xi_{-K}, \tau_{-K}, \ldots, \xi_{-1}, \tau_{-1})$ since the definition of K is not constructive (the state space is continuous) and because m may be too large to be amenable to any efficient computation.

Instantiating with Squirrel: In the Squirrel example, N follows a birth and death process. If a large number of consecutive departures occurs, then the number of customers becomes $N = 0$ and the number of files must also be null in that case: $x = 0$. Moreover, for all n, $\mathbb{P}(N(\Upsilon_n) = 0|N(0) = N_0) \geq \mathbb{P}(N(\Upsilon_n) = 0|N(0) = N_{max})$. Now, $\mathbb{P}(N(\Upsilon_n) = 0|N(0) = N_{max}) \geq \frac{\mu^n N_{max}!}{\Lambda^n}$ if $n \geq N_{max}$. It suffices to choose $m = N_{max}$, $h = \mathbb{1}_{(0,0)}$ and $B = (\mu/\Lambda)^{N_{max}} N_{max}!$ in the definition of uniform ergodicity to satisfy the condition of uniform ergodicity required in Theorem 2. Therefore, the Squirrel MC, $S_n = (N(\Upsilon_n), x(\Upsilon_n))$ admits a vertical backward coupling time.

4.4 Monotonicity issues

From now on, we assume that the domain $\mathcal{D}$ admits a component-wise ordering: $(N, x) \preceq (N', x')$ if $N \leq N'$ and $x \leq x'$. If the construction Φ of the chain $S_n = \Phi(S_{n-1}, \xi_n, \tau_n)$ is monotonic in its state coordinate, ($\forall y, z \quad \Phi(u, y, z) \leq \Phi(v, y, z)$ whenever $u \leq v$) then, one can characterize the backward coupling time in terms of maximal and minimal states.

Since Φ is a combination of several functions, Φ is non-decreasing if F, h and φ are all non-decreasing in their first two coordinates N and x. Finally note that F is monotonic in its first two coordinates if f is monotonic in N and the solution of the differential equation is monotonic in its initial condition.

Theorem 3 ([Foss and Tweedie, 1998]). *Let $\overline{M}$ (resp. $\underline{M}$) be the set of all maximal (resp. minimal) elements in $\mathcal{D}$ for the order $\preceq$. Then K' equal to*

$$\min\{n \geq 0 : \Phi(s, \xi_{-n}, \tau_{-n} \ldots, \xi_{-1}, \tau_{-1}) = \Phi(r, \xi_{-n}, \tau_{-n} \ldots, \xi_{-1}, \tau_{-1}), \forall r \in \overline{M}, s \in \underline{M}\},$$

is a vertical backward coupling time of the chain.

Note that if $\overline{M}$ and $\underline{M}$ are finite sets, then this definition of K' is constructive so that one can use backward coupling to design a perfect simulation algorithm of the Markov chain.

Instantiating with Squirrel: Monotonicity is a rather strong assumption for general hybrid systems as defined in Section 2. The Squirrel instance does not satisfy this assumption. We recall that the Squirrel Markov chain is such that :

1. if $\xi > 1 - \lambda'(N_{max} - N(\Upsilon_{n-1}))$, then the next event is a customer arrival: $N(\Upsilon_n) = N(\Upsilon_{n-1}) + 1$ and $x(\Upsilon_n) = x(\Upsilon_n^-)$;
2. if $\xi < \mu' N(\Upsilon_{n-1})$, then the next event is a customer departure $N(\Upsilon_n) = N(\Upsilon_{n-1}) - 1$ and $x(\Upsilon_n) = x(\Upsilon_n^-) - x(\Upsilon_n^-)/N(\Upsilon_{n-1})$;
3. otherwise, this is a null event, *i.e.* no customer arrives nor leaves and the system is left unchanged at time Υ_n.

As for the evolution between jumps, it follows the differential equation (2). To test if the chain is monotonic in its two components, one considers two chains S_1 and S_2 starting with ordered values, $N_u(0) \geq N_l(0)$ and $x_u(0) \geq x_l(0)$.

One must first consider the evolution between jumps. It should be clear that the differential equation (2) is monotonic in N as well as in its initial condition. Therefore, if $N_u(0) \geq N_l(0)$ and $x_u(0) \geq x_l(0)$, then for all time $t \geq 0$, $x_u(t) \geq x_l(t)$.

As for the behavior of the chain at jump times, it is monotonic as long as $\xi > 1 - \lambda' N_{max}$; this corresponds to arrivals or null events, whatever the value of N. In that case, if $N_u(\Upsilon_{n-1}) > N_l(\Upsilon_{n-1})$ then $N_u(\Upsilon_n) \geq N_u(\Upsilon_{n-1}) \geq N_l(\Upsilon_{n-1}) + 1 \geq N_l(\Upsilon_n)$ since it may be that ξ corresponds to an arrival for S_2 an a null event for N_u. As for the files, such events do not change their values: $x_u(\Upsilon_n) = x_u(\Upsilon_n^-) \geq x_l(\Upsilon_n) = x_l(\Upsilon_n^-)$.

When $\xi < 1 - \lambda' N_{max}$, the event may either be a departure or a null event for both chains. The following tricky situation can occur: $\mu' N_l(\Upsilon_{n-1}) < \xi < \mu' N_u(\Upsilon_{n-1})$. This corresponds to a departure for S_1 and a null event for S_2. Now, if $x_u(\Upsilon_n^-)$ and $x_l(\Upsilon_n^-)$ are too close, the following can happen: $x_u(\Upsilon_n) = x_u(\Upsilon_n^-) - x_u(\Upsilon_n^-)/N_u(\Upsilon_{n-1}) \leq x_l(\Upsilon_n^-) = x_l(\Upsilon_n)$. So that the chain is actually not monotonic under such events.

4.5 Stochastic envelopes of the Markov chain

To address this non-monotonicity problem we introduce upper and lower envelopes of the trajectories of the Markov chain S. These envelopes are constructed to keep containment along all trajectories. However, they are not Markovian in isolation.

In the following, we will consider that the order over the state space is a lattice and that $\overline{M}$ and $\underline{M}$ are reduced to single states. If this is not the case, the following can be adapted but becomes more technical. We define upper and a lower envelopes, $\mathcal{U} = (\overline{N_u}, \overline{x_u})$ and $\mathcal{L} = (\underline{N_l}, \underline{x_l})$, respectively, of all trajectories of the Markov chain, starting from all possible states in $\mathcal{D}$. The upper (resp. lower) envelope starts at time $t = 0$ in state $\overline{M}$ (resp. $\underline{M}$). They are both sampled on the underlying process $\{\Upsilon_n\}_{n\in\mathbb{N}}$ (so that some instants Υ_n may not correspond to actual jumps on one or both envelopes).

$$\begin{aligned}\mathcal{U}(\Upsilon_n) &= \sup_{s\in\mathcal{A}(n-1)} \Phi(s, \xi_n, \tau_n),\\ \mathcal{L}(\Upsilon_n) &= \inf_{s\in\mathcal{A}(n-1)} \Phi(s, \xi_n, \tau_n),\end{aligned}$$

where $\mathcal{A}(n-1)$ is the set of all states s such that $\mathcal{L}(\Upsilon_{n-1}) \leq s \leq \mathcal{U}(\Upsilon_{n-1})$. The upper envelope is not a Markov chain, neither is the lower one. However, the couple $(\mathcal{U}(\Upsilon_n), \mathcal{L}(\Upsilon_n))$ is a Markov chain over the continuous state space $\mathcal{D} \times \mathcal{D}$. The construction of the envelopes $(\mathcal{U}(\Upsilon_n), \mathcal{L}(\Upsilon_n))$ given above can be written under the form of two new functions Γ_1, Γ_2 that describes the Markovian evolution of both envelopes at jump times.

$$\mathcal{U}(\Upsilon_n) = \Gamma_1\Big(\mathcal{U}(\Upsilon_{n-1}), \mathcal{L}(\Upsilon_{n-1}), \xi_n, \tau_n\Big), \tag{9}$$

$$\mathcal{L}(\Upsilon_n) = \Gamma_2\Big(\mathcal{U}(\Upsilon_{n-1}), \mathcal{L}(\Upsilon_{n-1}), \xi_n, \tau_n\Big). \tag{10}$$

Note that by construction of the envelopes, $\mathcal{U}(t)$ stays above $\mathcal{L}(t)$ for all time $t \geq 0$ using the property $\mathcal{U}(0) > \mathcal{L}(0)$. Also note that by construction, for all initial state $S(0) = (N, x)$ and all time t,

$$\mathcal{L}(t) \preceq S(t) \preceq \mathcal{U}(t).$$

Theorem 4. *Assume that the Markov chain $(\mathcal{U}(\Upsilon_n), \mathcal{L}(\Upsilon_n))$ hits the diagonal (i.e., states of the form (s, s)) in finite time a.s.. The hitting time*

$$K'' = \min\big\{n : \Gamma_1\big((s, r, \xi_{-K}, \ldots, \xi_0, \tau_{-K}, \ldots, \tau_0\big) = \Gamma_2\big((s, r\xi_{-K}, \ldots, \xi_0, \tau_{-K}, \ldots, \tau_0\big)\forall s \in \overline{M}, r \in \underline{M}\big\}$$

is a vertical backward coupling time of the Markov chain S, so that for all initial state s,

$$\Phi(s, \xi_{-K''}, \tau_{-K''}, \ldots, \xi_{-1}, \tau_{-1}) \sim \Pi$$

.

Proof. It simply uses the fact that $\mathcal{U}(t) \geq S(t) \geq \mathcal{L}(t)$ for all initial conditions for the chain S. Consider a stationary initial condition $S(-K'') \sim \Pi$. Then, $S(0) = \Phi(S(-K''), \xi_{-K''}, \tau_{-K''}, \ldots, \xi_{-1}, \tau_{-1}) \sim \Pi$ by stationarity and $\mathcal{U}(0) = S(0) = \mathcal{L}(0)$ by definition of K''. □

Remark 5. *In general, this approach does not gain much over the general non-monotonous coupling techniques because of two problems:*

- *The assumption that $(\mathcal{U}(\Upsilon_n), \mathcal{L}(\Upsilon_n))$ hits the diagonal may not be verified,*
- *even if Theorem 4 applies, the time needed to compute $\mathcal{U}$ and $\mathcal{L}$ might be complicated and might not provide any gain over considering all states in $\mathcal{A}(n-1)$*

The goal of the next paragraph is to show that both obstacles can be overcome in the Squirrel case.

Instantiating with Squirrel: The upper (resp. lower) envelope start in $(N_{\max}, C)$ (resp. $(0,0)$) and evolves exactly as the Markov chain over all events that cannot cause a swap of the ordering between the two envelopes. Whenever such an event occurs, here is the way both envelopes are computed in constant time. Let $N_3 = \lfloor \xi/\mu' \rfloor$ be the largest value of N for which ξ is the null event. For N_3+1 and larger values of N, ξ would be a departure. Note that since ξ is a swapping event, then $\overline{N_u}(\Upsilon_{n-1}) > N_3 \geq \overline{N_l}(\Upsilon_{n-1})$. Now, the smallest value of N after event ξ is larger than $\underline{N_l}(\Upsilon_{n-1})$, the smallest value of x is larger than $\underline{x_l}(\Upsilon_n^-)\frac{N_3}{N_3+1}$, while the largest possible value of N after event ξ is smaller than $\overline{N_u}(\Upsilon_{n-1})-1$ and the largest possible value of x after event ξ is smaller than $\overline{x}_1$. Therefore, we set $\mathcal{U}(\Upsilon_n) = (\overline{N_u}(\Upsilon_{n-1}) - 1, \overline{x}_1)$ and $\mathcal{L}(\Upsilon_n) = (\underline{N_l}(\Upsilon_{n-1}), \underline{x_l}(\Upsilon_n^-)\frac{N_3}{N_3+1})$. Therefore, an event that would correspond to a departure in $\mathcal{U}$ and a null event in $\mathcal{L}$ becomes a "dummy" departure for $\mathcal{L}$ and a "dummy" arrival in $\mathcal{U}$.

Lemma 6. *For the Squirrel Markov chain, $(\mathcal{U}(\Upsilon_n), \mathcal{L}(\Upsilon_n))$ hits the diagonal (more precisely, the Squirrel Markov chain hits the state $(0,0)$) in finite time a.s.*

Proof. The proof is similar to the proof of the uniform ergodicity of chain S. Indeed, if a large number of departures occur, both envelopes will eventually reach $(0,0)$. Since this happens with a positive probability, the Markov chain $(\mathcal{U}(\Upsilon_{n-1}), \mathcal{L}(\Upsilon_{n-1}))$ is uniformly ergodic and K'' has a finite expectation. □

4.6 Perfect Simulation Algorithm for hybrid models

The previous theorem 4 is the theoretical foundation of the perfect simulation algorithm using the above envelopes.

Algorithm 1 provides the perfect sampling algorithm using Theorem 4. There are two subtleties added to the "coupling from the past" technique.

First, Algorithm 1 uses a time doubling technique for reducing the total computation time, as suggested in [Propp and Wilson, 1996]. Its time complexity is $O((2K + c_\Phi)(|MAX| + |MIN|))$, where c_Φ is the total complexity of computing Φ over the whole simulation.

Second, the stopping test is relaxed. Here we decide to stop as soon as the upper and lower envelopes are close enough. A stopping test on equality is theoretically possible since both envelopes couple in $(0, 0)$ with positive probability, and remain exactly equal from this point on. However, the probability that N ever reaches 0 is extremely small (less than 10^{-300} in the examples of Section 5). This means that the average vertical coupling time is huge so that the backward coupling technique of no practical use. Testing for a small gap between both envelopes provides a strong control on the simulation time (see Section 6). It also provides confidence intervals on the output of the simulation, as seen in the following.

Algorithm 1 Backward simulation of a hybrid system using envelopes

$n = 1$;
repeat
  $\mathcal{U} := (N_{max}, C)$,$\mathcal{L} := (0, 0)$ {Initialize the two envelopes at time $-n$}
  for $i = n$ downto $\lfloor n/2 \rfloor + 1$ **do**
    $\xi[-i]$:=Random(Uniform over $[0, 1]$); $\tau[-i]$:=Random(exponential with rate Λ) {generates the event at step $-i$ and the inter-jump time}
  end for
  for $i = n$ downto 1 **do**
    $\mathcal{U} := \Gamma_1(\mathcal{U}, \mathcal{L}, \xi[-i], \tau[-i])$, $\mathcal{L} := \Gamma_2(\mathcal{U}, \mathcal{L}, \xi[-i], \tau[-i])$
  end for
  $n = 2n$;
until $(\mathcal{U} \to N) = (\mathcal{L} \to N)$ and $(\mathcal{U} \to x) - (\mathcal{L} \to x) \leq \varepsilon/3$
return $\mathcal{U}, \mathcal{L}$

4.6.1 Stationary state at arbitrary instants

Algorithm 1 provides samples of the stationary distribution at jump instants. However, this distribution may significantly differ from the distribution of the system at arbitrary instants. From the PASTA [Baccelli and Brémaud, 1994] property, this latter distribution can be sampled simply by pursuing each trajectory during a random time independent of the system, distributed according to the jump process. More precisely, the algorithm gives a trajectory which is in stationary state at time 0. Therefore, this trajectory will remain stationary ever after 0. To observe an arbitrary stationary sample, on can thus choose an arbitrary instant after time 0. This arbitrary instant can for instance be sampled according to an exponential distribution with

rate equal to the Poisson arrival rate, since the PASTA property ensures that such a sampling is representative of the system at arbitrary instants (not necessarily jump instants).

Another possibility is to use backward sampling to obtain one stationary sample at time 0 (given by algorithm 1) and then use this state as an initial state for a classical forward simulation. The stationary behavior is obtained by sampling the visited states over this single trajectory from time 0 to some large simulation time T, using a deterministic periodic sampling sequence. The ergodicity property of the system ensures that this technique provides samples distributed according to the general stationary distribution when T goes to infinity.

4.6.2 Implementation issues

The perfect simulation theorem of Propp and Wilson for finite Markov chain can be almost directly translated into an implemented algorithm for sampling the stationary distribution of the chain. This is not really the case for Theorem 2 nor for Theorem 3 for two reasons. First, the vertical coupling times may be too large for a computer program to ever converge in a reasonable time. Second, the definition of the vertical coupling times is based on testing the equality of real numbers, which is always difficult to do in a computer program.

Both issues can be addressed by choosing an arbitrary precision ε and by stopping the algorithm as soon as all the trajectories are contained in a ball of size ε.

The output of the algorithm, in the monotonic version is an upper and a lower bound on the cumulative distribution function of Π, with precision ε.

In the experiments provided below (Section 5), this method improves the convergence time by several orders of magnitude, even when ε is very small. An explanation of this behavior is also provided.

Instantiating with Squirrel: In Figure 4, we display a perfect simulation of $\mathcal{U}, \mathcal{L}$ together with an arbitrary trajectory of the Squirrel Markov chain S, starting in $N_{max}/2, C/2$. First note that S stays within $[\mathcal{L},\mathcal{U}]$ at all times. Also note that several dummy events on $\mathcal{U}, \mathcal{L}$ are visible. They all correspond to discontinuous jumps of $\mathcal{L}$ and singularities of $\mathcal{U}$. At those instants, S may or may not jump, depending on the value of ξ.

Algorithm 1 is used to simulate the hybrid squirrel model. The outputs of the i-th run of the algorithm are numerical approximations of the couples $(\mathcal{U}^i = (N_u^i, x_u^i), \mathcal{L}^i = (N_l^i, x_l^i))$. Using a small step h for numerical integration of the differential equations (*i.e.* such that the error $\epsilon = h\sigma N + \alpha\tau \frac{\sigma^{\alpha+1} g N^{\alpha+1}}{C^\alpha} h^\alpha \leq \varepsilon/3$, see Appendix A.5), then the outputs of the algorithm are such that the real values $x_u^i - x_l^i \leq \varepsilon$.

Let us denote by π_N and π_x the marginal distribution of N and x respectively, for the unknown

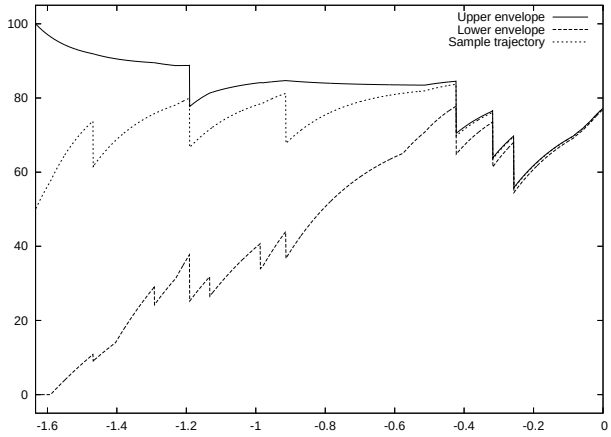

Figure 4: Trajectories of the two envelopes $\mathcal{U}, \mathcal{L}$ and a trajectory of the Markov chain S. One can notice several "dummy" departures on S_2 and several "dummy" arrivals on S_1. The sample of S is a real trajectory, which stays within the upper and lower envelopes.

distribution Π. According to Theorem 4, they satisfy for all runs i:

$$P(N_u^i = k) = P(N_l^i = k) = \pi_N(k), \tag{11}$$

$$P(x_u^i \leq z | N_u^i) \leq \pi_x([0,z]|N_u^i) \leq P(x_l^i \leq z | N_l^i) \leq P(x_u^i - \varepsilon \leq z | N_u^i) = P(x_u^i \leq z + \varepsilon | N_u^i). \tag{12}$$

From these equations, one can compute confidence intervals using the following procedure. Let E be an interval $E = [a, b] \subset [0, C]$ for which we want to compute $\pi_x(E)$ with confidence c. The problem is to find an interval I such that $P(\pi_x(E) \in I) \geq c$.

We denote by E_ε the augmented interval , $E_\varepsilon \stackrel{\text{def}}{=} [a - \varepsilon, b + \varepsilon]$. For a total of M runs, let $\hat{p_1} = 1/M \sum_{i=1}^{M} \mathbf{1}\{x_u^i \in E \wedge x_l^i \in E\}$ and $\hat{p_2} = 1/M \sum_{i=1}^{M} \mathbf{1}\{x_u^i \in E_\varepsilon \vee x_l^i \in E_\varepsilon\}$. Using Equation (12), There exists an unbiased estimator $\hat{p}$ of $\pi_x(E)$ such that $\hat{p_1} \leq \hat{p} \leq \hat{p_2}$.

Now, the central limit theorem gives the following confidence interval

$$I = \left[\hat{p_1} - \frac{\eta_c v}{\sqrt{M}}, \hat{p_2} + \frac{\eta_c v}{\sqrt{M}} \right],$$

where η_c is half the c-percentile for the normal distribution $\mathcal{N}(0,1)$ and v is the variance of a Bernoulli distribution with probability $\pi_x(E)$: $v = \sqrt{\pi_x(E)(1 - \pi_x(E))} \leq 1/2$.

Note that the size of the confidence interval can be bounded by $\frac{\eta_c}{\sqrt{M}} + e$ where the error e is made by substituting $\hat{p}$ by the empirical measures $\hat{p_1}$ and $\hat{p_2}$. If we denote the cumulative distributions $\pi_1([u,v]) \stackrel{\text{def}}{=} P(u \leq x_u^i \leq v | N_u^i)$, $\pi_2([u,v]) \stackrel{\text{def}}{=} P(u \leq x_l^i \leq v | N_l^i)$, that do not depend on i, then

$$e \leq \sup_{z>0} \pi_1([z, z+\varepsilon]) + \sup_{z>0} \pi_2([z, z+\varepsilon]).$$

It should be noted that since the only atom of all the distributions is in 0 and the supremum is taken over all $z > 0$, The error e goes to 0 when ε goes to 0.

So the size of the confidence interval decreases with $1/\sqrt{M}$, as usual with unbiased estimates and with the error e which is an empirical function of the precision ε.

5 Numerical experiments

In this section we report the results of several simulations made on the Squirrel model. We report the computation times as well as the performance indexes measured over the system. All experiments are carried out on a 2 GHz Pentium 4 with 1 GB of RAM.

We choose a realistic instance of the Squirrel model with

- $C = 10000$, which is the maximal number of files that can be present simultaneously over the system;
- $N_{max} = 1000$, which is the maximal number of participants to our Squirrel system;
- $\mu = \lambda = 10^{-4}$ (on average, each customer stays connected/disconnected for 2.7 hours);
- $\sigma = 10^{-2}$ (on average, each connected customer emits a request every 1.6 minutes);
- $\theta = 10^{-2}$, corresponds to an 1.6 minutes time-to-live;
- $\alpha = 0.4$, corresponds to a Zipf-like popularity distribution with parameter 0.8.

On Figures 5 and 6 we show the density of N and x, respectively. To obtain these figures we ran algorithm 1 or equivalently the "N-first" algorithm (see section 6) 10000 times in approximately 4 minutes and 30 seconds. The ten thousand resulting samples of N and x are distributed according to the stationary distribution. On Figure 5 we observe the density of the Ehrenfest model, which is highly centered around its mean value. On Figure 6 we observe that the distribution of x also exhibit a very narrow peak around its mean value. This shows that x has very little variance and is well described by its mean. For this reason in the following experiment we restrict our attention to the mean value of x.

We then chose to measure H, the asymptotic *hit probability*. This is the probability that a given file (chosen uniformly over all files) is present in the system, under its stationary regime. As shown in Appendix A.1,

$$H = \lim_{t \to \infty} \left(\frac{x(t)}{C} \right)^{\alpha}.$$

This quantity is the typical performance measure of a caching system like Squirrel.

Figure 6 shows that for the Squirrel system described above, there are on average 500 simultaneously connected nodes and the average number of cached documents is 363.78, corresponding to an average hit probability of 0.26.

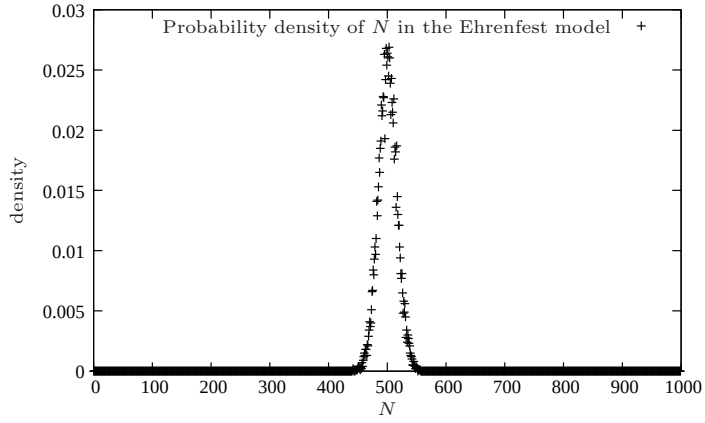

Figure 5: The empirical density of the distribution of N, as given by 10000 simulations.

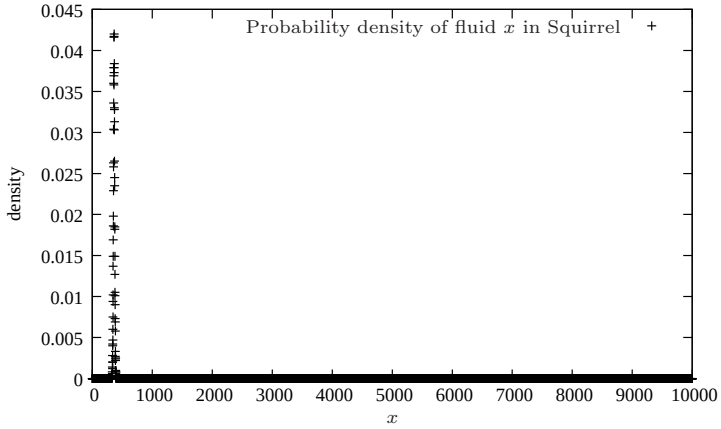

Figure 6: The empirical density of the distribution of x, as given by 10000 simulations.

We now show how the hit probability varies with regard to its design parameters. Note that θ is the only parameter over which the system designer has some control (the time-to-live of cached objects may be set to an optimal default value). Figure 7 displays the hit probability (and not the amount of fluid x) as θ varies from 10^{-7} to 10 for the simulation setup described above.

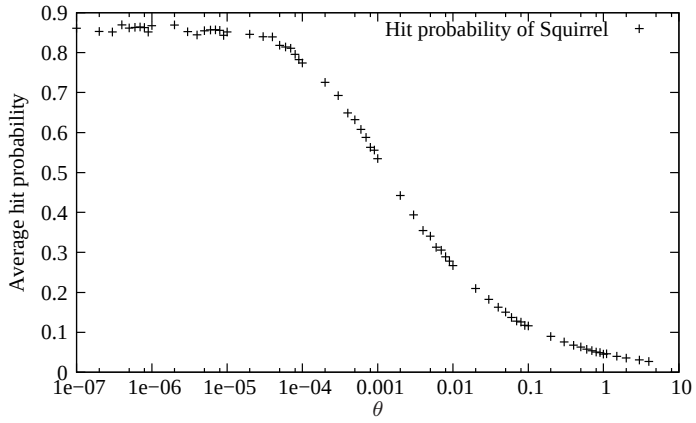

Figure 7: The hit probability as θ varies (over a log scale).

Note that θ evolves over a logarithmic scale. We observe that the hit probability can vary greatly with θ (from 0 to 0.9). However, only the $[10^{-4}, 10^{-2}]$ range of θ has a real impact on the hit probability: the hit probability drops sharply as soon as $\theta > 10^{-4}$ and is below 0.2 as soon as $\theta > 10^{-2}$. A 0.5 hit probability is achieved with $\theta > 10^{-3}$. This means that for the system described above, the time-to-live of objects should not be of the order of 15 minutes. We also note that the smaller θ the better hit probability. However, a small θ means a increased risk of using out-of-date cached copies. To limit this drawback most caching systems use a default 24 hours time-to-live corresponding to $\theta \approx 10^{-5}$. This values seems to be a minimum value to ensure acceptable average staleness.

6 Coupling time improvements in Squirrel

The most important feature of perfect simulation is the coupling time. Indeed, perfect simulation is not usable for sampling the stationary distribution if the coupling time is too large. Two factors play a major role in the coupling time.

The first factor is the convergence rate of the stochastic discrete part, N. Since, the underlying Markov Chain is the Ehrenfest model, the stationary distribution is not spread and both the upper and lower envelopes have their respective discrete part converge fast towards middle values. The second factor is the ratio between the convergence rate of the differential equation to its asymptotic value and the jump rate of the discrete part. The next two sections address these two points.

6.1 Let N converge first

One can take advantage of the fact that the evolution of N is independent of the evolution of x, so that its stationary distribution can be computed in isolation.

The idea is to first compute a sample N_∞ for N, distributed according to its marginal stationary measure, and then to simulate for the global system (N, x) starting with the initial states (N_∞, C) and $(N_\infty, 0)$, instead of (N_{max}, C) and $(0, 0)$. This is valid using the next theorem.

Theorem 7. *If N_∞ is a sample of N with the stationary distribution of the Markov chain $N(\Upsilon_n)$, then,*

$$(N_0, x_0) = \Phi\Big((N_\infty, C), \xi_{-K'''}, \tau_{-K'''}, \ldots, \xi_{-1}, \tau_{-1}\Big),$$

is a sample with the stationary distribution of the Markov chain $(N(\Upsilon_n), x(\Upsilon_n))$ if K''' is the vertical backward coupling time of the trajectories starting with the initial states (N_∞, C) and $(N_\infty, 0)$:

$$K''' \stackrel{\text{def}}{=} \min\{n : \Phi\Big((N_\infty, C), \xi_{-n}, \tau_{-n}, \ldots, \xi_{-1}, \tau_{-1}\Big) = \Phi\Big((N_\infty, 0), \xi_{-n}, \tau_{-1}, \ldots, \xi_{-1}, \tau_{-1}\Big).$$

Proof. Starting from a stationary state (N_∞, x_∞) at time $-K'''$, the state at time 0 must be (N_0, x_0) by monotonicity of the chain with respect to x. The state at time $-K'''$ being stationary, so is the state at time 0, therefore, (N_0, x_0) has the stationary distribution of the chain $(N(\Upsilon_n), x(\Upsilon_n))$. □

This construction has several advantages.

First, N_∞ can be constructed without using a simulation since the stationary distribution of the Ehrenfest birth and death process $N(\Upsilon_n)$ is well known (see Equation (8)). Generating a sample according to this distribution is rather simple and can be done in constant time using an aliasing technique [Walker, 1974].

A second advantage is that since both trajectories always have the same value for N, the associated Markov chain becomes monotonic and there is no need to use envelopes. This improves the coupling time since dummy events that tend to widen the gap between the upper and lower trajectories never occur here.

We have used this approach and have evaluated its benefit in terms of coupling time as well as total computation time. In the following experiment, we ran both algorithm 1 and this N_∞ enhancement. The experimental setup is the following. We chose the realistic parameters $\alpha = 0.4, \theta = 10^{-4}, \sigma = 10^{-3}, C = 10000, N_{max} = 5000, \lambda = \mu = 10^{-4}$. For each experiment using this setup, the coupling time was 65536 (2^{16}) steps for algorithm 1 and 32768 (2^{15}) for the N_∞ algorithm (note that the coupling time is necessarily a power of two because of the time doubling technique). Regarding the real execution time (measured with the *date* command), the N_∞ algorithm takes less than 45 minutes (for 1000 runs) while the original algorithm 1 ended after more than 2 hours

and 40 minutes, over a Pentium 4 PC. The fact that the N_∞ algorithm is more than twice faster than the original algorithm while its coupling time is just halved, comes from the fact that there is no need to generate envelopes in that case. This saves time in the inner loop of the algorithm.

Finally, this approach provides a better understanding on the simulation duration. When starting from states (N_∞, C) and $(N_\infty, 0)$, the rates of convergence over both trajectories (see Lemma 11 in Appendix A) is uniformly bounded by $\gamma \stackrel{\text{def}}{=} \max_{N=1}^{N_{max}} \gamma_N$.

From Lemma 11, the distances $|x_u(t)-\ell_N|$ and $|x_l(t)-\ell_N|$ are both bounded by $e^{-\gamma t}$ over all the inter-jump intervals. Therefore, $|x_u(t) - x_l(t)| \leq 2Ce^{-\gamma t}$. Note that this bound does not depend on N anymore. Also note that $|x_u(t) - x_l(t)|$ never increases at jump times: $|x_u(T_n) - x_l(T_n)| \leq |x_u(T_n^-) - x_l(T_n^-)|$.

The duration before coupling T verifies $|x_u(T) - x_l(T)| \leq \varepsilon/3$, so that

$$T \leq -\gamma \log(\varepsilon/6C) = \gamma(\log(6C) - \log(\varepsilon)). \tag{13}$$

This bound is rather tight since the rate of convergence is never much larger than γ, over the whole range of N and x.

As for the vertical coupling time K''' of the simulation, K''' corresponds to a sum of independent exponential variables with rate Λ such that $\sum_{i=1}^{K'''} \tau_i \geq T \geq \sum_{i=1}^{K'''-1} \tau_i$. Therefore, $K''' - 1$ has a Poisson distribution with rate ΛT. Its mean verifies $\mathbb{E}K''' \leq -\Lambda\gamma \log(\varepsilon/6C) + 1$ and its variance $varK''' \leq -\Lambda\gamma \log(\varepsilon/6C) + 1$.

This explains why most simulations have the same number of steps (which is a power of 2, such that $2^{n-1} \leq K''' \leq 2^n$), since T is almost a deterministic value.

Theorem 8. *The time complexity of the simulation algorithm using N_∞ is*

$$O\left(N_{max}(\log C - \log \varepsilon)\left(1 + \frac{(N \log(C/\varepsilon))^{1/\alpha}}{C\varepsilon^{1/\alpha}}\right)\right),$$

and its space complexity is $O(N_{max}(\log C - \log \varepsilon))$.

Proof. The complexity of the algorithm is $O((K''' + c_\Phi)(|MAX| + |MIN|))$ where K''' is the vertical coupling time, c_Φ is the total cost of computing Φ and $(|MAX| + |MIN|)$ is the number of simulations to be run in parallel. Here , $(|MAX| + |MIN|) = 2$ and we have shown above that $K''' = O(N_{max}(log(C) - log(\varepsilon))$. As for c_Φ, it is proportional to the number of steps for integrating the differential equation over time T. If each step is of size h such that $h\sigma N + \alpha\tau \frac{\sigma^{\alpha+1} g N^{\alpha+1}}{C^\alpha} h^\alpha \leq \varepsilon/3$ (see Appendix A.5), then $c_\Phi = O(T/h) = O(\frac{N^{1+1/\alpha}}{C\varepsilon^\alpha}(\log C - \log \varepsilon)^{1+1/\alpha}$. Therefore, the total complexity is $O\left(N_{max}(\log C - \log \varepsilon)\left(1 + \frac{(N \log(C/\varepsilon))^{1/\alpha}}{C\varepsilon^{1/\alpha}}\right)\right)$.

The space complexity comes from the fact that the random innovations ξ_n and τ_n need to be stored for the total duration K of the simulation. □

It is quite remarkable that the complexity is linear in N_{max} and in $\log C$, when C is of order N^2 and $\alpha > 1/2$, which is typical in Squirrel systems.

6.2 Asymptotic solutions

The solutions of the differential equations converge to a unique asymptotic value, ℓ_N if no jumps ever occur (or equivalently, if N remains constant) (see Lemma 9 in Appendix A for more on this). This is the number of copies present in the system on average when N customers are present and when nobody ever leaves or joins in.

If the rate of convergence γ_N of x towards its asymptotic value is larger than the jump rate Λ, then in most cases, x will actually be very close to ℓ_N before the next change occurs. Therefore, one may disregard the transient behavior of x and let x jump from ℓ_N to $\ell_{N'}$ whenever a jump from N to N' occurs. This actually makes the system discrete since both x and N only take discrete values.

Computing the asymptotic value, ℓ_N is much faster than solving the differential equation numerically, and can be done by solving numerically in x (using Newton's method).

$$\sigma N \left(1 - \left(\frac{x}{C}\right)^{\alpha}\right) - \theta x = 0. \tag{14}$$

In that case, the stationary distribution of (N, x) can be approximated by

$$\Pi'(N = k, x = \ell_k) = \frac{\left(\frac{\lambda}{\mu}\right)^k \binom{N_{max}}{k}}{\left(1 + \frac{\lambda}{\mu}\right)^{N_{max}}}.$$

Generating samples according to this distribution is rather simple and can be done in constant time using aliasing techniques.

We have compared this approximation with the exact samples (N_u, x_u) computed by the simulation algorithm presented in Section 4 (using the absolute value $|\ell_{N_u} - x_u|$). As seen in Figure 8, the approximation using the asymptote is very good as soon as γ_{N_u}, the rate of convergence of the solution of the differential equation to its asymptote it larger than 5% of the jump rate Λ.

As a rule of thumb, for large squirrel systems (with a large number of customers (N_{max}) and a very large file system (C)), the asymptotic approximation is valid as soon as the inverse of the life time of files , θ is of the same order as $C(\lambda + \mu)/N_{max}$.

7 Conclusion

In this paper we have presented a general simulation framework for stochastic hybrid systems providing guaranteed samples. We demonstrated the applicability of this technique by carrying out an experimental study of a complex peer to peer system modeled by hybrid equations.

Simulations based on backward coupling techniques are particularly well adapted to the case of stochastic hybrid systems for several reasons.

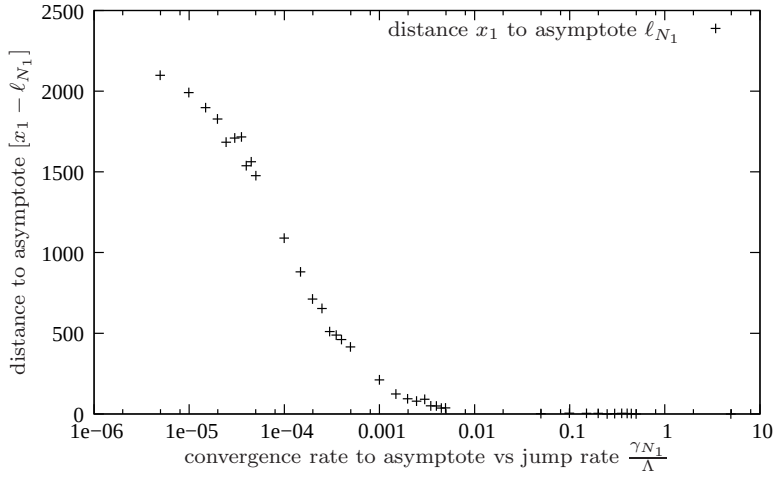

Figure 8: Comparing the samples of perfect simulations with their asymptotic approximation.

- They provide numerical guarantees,
- and they get rid of the difficulty to manipulate continuous variables.
- Finally, hybrid systems often mean large state spaces. These large state spaces often make numerical computations of the behavior of the system impossible. However, the complexity of our backward coupling depends on the size of the state space in a sub-linear way. It is remarkable that in the Squirrel example, the complexity is in $O(N_{max}(\log(C) - \log(\varepsilon))$ in most realistic cases, while the state space is $N_{max}C$.

Acknowledgements

We would like to thank the three reviewers of this paper whose in-depth study of the paper and constructive comments helped us greatly in improving the quality of this paper.

A More on the Squirrel differential equation

The appendix is used to explain the construction of the differential equation

$$\frac{dx}{dt} = f_N(x(t)) = \sigma N \left(1 - \left(\frac{x(t)}{C}\right)^{\alpha}\right) - \theta x(t), \tag{15}$$

and to assess its properties (uniqueness of the solution, asymptotic behavior, rate of convergence).

A.1 Modeling the miss probability

We now explain how the miss probability of Squirrel can be modeled by

$$1 - \left(\frac{x}{C}\right)^{\alpha}. \tag{16}$$

This model was first proposed in [Clévenot and Nain, 2004, Clévenot-Perronnin, 2005]. In this section we will not only recall the theoretical grounds for this model, but we will also provide experimental results to validate the model and estimate α.

A uniform-popularity model would straightforwardly give a linear model $P(\text{hit}|N(t), x(t)) = \frac{x(t)}{C}$ (the hit probability being 1 - the miss probability). However, it was shown in [Breslau et al., 1999] that Web objects exhibit a Zipf-like popularity distribution, i.e., the k-th most popular object is requested with probability $\frac{\Omega}{k^\beta}$, where $0 < \beta < 1$ is the skew parameter of the distribution and $\Omega = 1/\sum_{i=1}^{C} i^{-\beta}$ is a normalization factor. In [Breslau et al., 1999] it was shown that β is close to 0.8. Assuming that the $x(t)$ present objects in cache are the $x(t)$ most popular ones (since they are more likely to be requested), a rough approximation for the hit probability is

$$P(\text{hit}|x(t)) = \sum_{i=1}^{\lfloor x(t) \rfloor} \frac{\Omega}{i^\beta} \approx \frac{x(t)^{1-\beta} - 1}{C^{1-\beta} - 1}, \tag{17}$$

by using the approximation $\sum_{i=1}^{\lfloor x \rfloor} i^{-\beta} \approx \int_1^x t^{-\beta} dt = (x^{1-\beta} - 1)/(1-\beta)$ for $x \geq 1$. From (17) we get the rough approximation $P(\text{hit}|x(t)) = (\frac{x(t)}{C})^\alpha$ with $\alpha = 1 - \beta$.

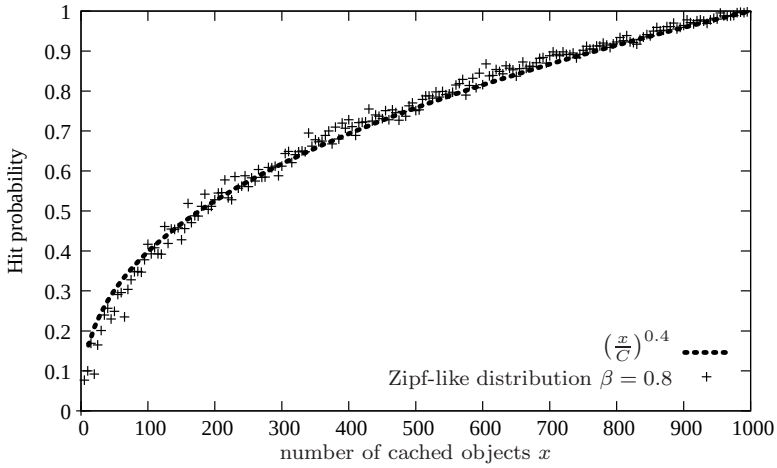

Figure 9: Hit probability as a function of the number of cached objects x for the Zipf-like distribution with parameter $\beta = 0.8$ and with $C = 1000$ possible objects.

To validate this model we have conducted the following experiment. We have initially drawn x samples from a Zipf-like distribution. Then we drew an additional sample z. This sample z results in a hit if it belongs to the set of x initially drawn samples, in a miss otherwise. This latter step is repeated 1000 times to get a 95% confidence interval no larger than 0.03 on the hit probability. Then we compared the experimental hit probability to the model given by (16) and we computed the value of α that best fitted the numerical results. Figure 9 show how perfectly (16) follows the experimental hit probability.

However, the value of α used to get this accuracy slightly differs from $1 - \beta$. Actually for most

values of β between 0.1 and 0.9 we found experimentally that $\alpha \approx 1.2 - \beta$ for $\beta < 1$. We conjecture that this discrepancy is partly due to the fact that the x cached objects are not necessarily the x most popular ones. It may also be due to the approximations used to get (17).

A.2 Asymptote

Lemma 9. *Any solution $x(t)$ of the differential equation converges to a unique limit ℓ_N, for all initial conditions.*

Proof. Setting $\frac{dx}{dt} = 0$ in (15) we get $f_N(x) = 0$. This equation is verified by potential asymptotes for $x(t)$.

Since $\frac{df_N}{dx} < 0$ for all $x \geq 0$ and for all $N > 0$, and since $f_N(0) = \sigma\, N > 0$ and $f_N(C) = -\theta\, C < 0$, equation $f_N(x) = 0$ admits a unique positive solution denoted by ℓ_N:

$$f_N(\ell_N) = 0. \tag{18}$$

Therefore any solution $x_N(t)$ of (15) admits a unique asymptote ℓ_N. □

Computing ℓ_N can be done by using a classical Newton method [Press et al., 1992], that converges very fast here.

A.3 Uniqueness of the solution

The differential equation (15) is not Lipschitz in 0 as soon as $\alpha < 1$. Therefore, the classical techniques used to prove uniqueness do not apply here. However, uniqueness is required for our simulation method to work.

Lemma 10. *The equation (15) admits a unique solution for any initial condition $x^0 \stackrel{\text{def}}{=} x(0) \geq 0$.*

Proof. Let us consider first the case $x^0 < \ell_N$. In that case, if $x_1(t)$ is one solution, $\frac{dx_1}{dt} = f_N(x_1(t)) > 0$ for all $t \geq 0$, so that $x_1(t)$ is strictly increasing from x^0 to ℓ_N. Therefore, the inverse function $x_1^{-1}(u) : [x^0, \ell_N) \to \mathbb{R}$ is well defined. If we consider another solution $x_2(t)$ then the derivative of the inverse functions $x_1^{-1}(u)$ and $x_2^{-1}(u)$ verify $\frac{dx_1^{-1}}{du}(u) = \frac{1}{f_N(u)} = \frac{dx_2^{-1}}{du}(u)$ for all $u \in [x^0, \ell_N)$. So that $x_1^{-1}(u) - x_2^{-1}(u)$ is constant and does not depend on u. When $u = x^0$, then $x_1^{-1}(x^0) = 0$ and $x_2^{-1}(x^0) = 0$. This implies that $x_1^{-1}(u) = x_2^{-1}(u)$ for all $u \in [x^0, \ell_N)$, so that $x_1(t) = x_2(t)$ for all $t \geq 0$.

If $x^0 = \ell_N$, then the derivative of any solution remains equal to zero, so that there is a unique solution $x(t) \equiv \ell_N$.

Finally, the case $x^0 > \ell_N$ is similar to the case $x^0 < \ell_N$, with decreasing solutions instead of increasing ones. □

A.4 Rate of convergence

Let us define the *rate of convergence* $\gamma_N(t)$ at time t of the ODE solution $x_N(t)$ as follows: if $x_N(t) < \ell_N$,

$$\gamma_N(t) \stackrel{\text{def}}{=} \lim_{d\to 0} \frac{(x_N(t+d) - \ell_N) - (x_N(t) - \ell_N)}{d(\ell_N - x_N(t))}, \tag{19}$$

and $\gamma_N \stackrel{\text{def}}{=} \lim_{d\to 0} \frac{(x_N(t+dt)-\ell_N)-(x_N(t)-\ell_N)}{dt(x_N(t)-\ell_N)}$ otherwise. Note that $\gamma_N > 0$ since $x_n(t) \leq \ell_N$ in steady state as shown in Appendix B.

We now simplify (19) when $x_N(t) < \ell_N$. The case $x_N(t) > \ell_N$ is exactly similar and will be omitted.

Taking the limit as $d \to 0$ we get

$$\begin{aligned}
\gamma_N(t) &= \lim_{d\to 0} \frac{x_N(t+d) - x_N(t)}{d(\ell_N - x_N(t))}, &(20)\\
&= \frac{\frac{dx_N}{dt}(t)}{\ell_N - x_N(t)}, &(21)\\
&= \frac{\frac{dx_N}{dt}(t) - f_N(\ell_N)}{\ell_N - x_N(t)}, &(22)
\end{aligned}$$

where (22) is obtained using (18). Then when $x_N(t) \to \ell_N$ we get the asymptotic rate of convergence,

$$\begin{aligned}
\gamma_N &= \lim_{x\to \ell_N} \frac{f_N(x) - f_N(\ell_N)}{\ell_N - x}, &(23)\\
&= -\frac{df}{dx}(\ell_N) = \frac{\sigma\alpha N + \theta(1-\alpha)\ell_N}{\ell_N}. &(24)
\end{aligned}$$

Lemma 11. *If $x_N(0) < \ell_N$, then for all $t > 0$, $\gamma_N(t) \geq \gamma_N$ and for all $t > 0$, $x_N(t) \geq \ell_N(1 - \exp(-\gamma_N t))$.*

Proof. Note that if $x_N(0) < \ell_N$ then (in the absence of jumps) we have $x_N(t) < \ell_N$ for all $t > 0$. From (21) and (24) we have for all N:

$$\begin{aligned}
\gamma_N(t) - \gamma_N &= \frac{\frac{dx_N}{dt}(t)}{\ell_N - x_N(t)} + \frac{df}{dx}(\ell_N),\\
&= \frac{f_N(x_N(t)) + (\ell_N - x_N(t))\frac{df}{dx}(\ell_N)}{\ell_N - x_N(t)},\\
&= \frac{\sigma N\left(1 - \left(\frac{x_N(t)}{C}\right)^\alpha\right) - \theta x_N(t) - (\ell_N - x_N(t))(-\frac{\sigma N\alpha(\frac{\ell_N}{C})^\alpha}{\ell_N} - \theta)}{\ell_N - x_N(t)}.
\end{aligned}$$

The numerator $g(x(t)) \stackrel{\text{def}}{=} \sigma N\left(1 - \left(\frac{x_N(t)}{C}\right)^\alpha\right) - \theta x_N(t) - (\ell_N - x_N(t))(-\frac{\sigma N\alpha(\frac{\ell_N}{C})^\alpha}{\ell_N} - \theta)$ is positive since:

$$\frac{dg}{dx}(x(t)) = \frac{\sigma N\alpha\ell_N^\alpha x_N(t)^\alpha(-\ell_N^{1-\alpha} + x_N(t)^{1-\alpha})}{C^\alpha x_N(t)\ell_N} < 0,$$

and $\lim_{t\to\infty} g(x(t)) = 0$. This proves the first inequality of the lemma.

Using (21) this first inequality gives

$$\begin{aligned}\frac{\frac{dx_N}{dt}(t)}{\ell_N - x_N(t)} &\geq \gamma_N \\ \frac{dx_N}{dt}(t) &\geq \gamma_N(\ell_N - x_N(t)).\end{aligned}$$

Let $y(t)$ be the solution of the first-order linear ODE $\frac{dy}{dt}(t) = \gamma_N(\ell_N - y(t))$ with $y(0) = 0$. This solution is $y(t) = \ell_N(1 - \exp(-\gamma_N t))$. We now prove the second inequality, i.e., that for all $t > 0$, $x_N(t) \geq y(t)$. Let us consider the derivative of the inverse functions (which are always well defined here):

$$\frac{dy^{-1}}{du}(u) = \frac{1}{\gamma_N(\ell_N - u)} \geq \frac{dx_N^{-1}}{du}(u).$$

As a result we have $x_N^{-1}(u) \leq y^{-1}(u)$ for all $u \geq 0$ since $x_N^{-1}(0) = y^{-1}(0) = 0$. This implies that $x_N(t) \geq y(t)$ for all $t \geq 0$. □

A.5 Numerical integration

First, note that in the case $\alpha = 1$ the ODE (15) admits a closed-form solution on $[T_n, T_{n+1})$:

$$x(t) = \frac{\sigma N(T_n)}{\frac{\sigma N(T_n)}{C} + \theta} + \left(x(T_n) - \frac{\sigma N(T_n)}{\frac{\sigma N(T_n)}{C} + \theta} \right) e^{-(t-T_n)(\theta + \frac{\sigma N(T_n)}{C})}. \tag{25}$$

However, when $\alpha \neq 1$, the equation does not have a closed form solution and must be solved numerically. In this paper, we only consider a first order Runge-Kutta method. More sophisticated integration is possible, yielding a better precision with fewer discretization steps.

The fact that the equation may not be Lipschitz in 0, makes the computation of the error a little tricky. Here, we simply compute a bound on the error made by using the classical Euler integration method: $x_n = x_{n-1} + h f_N(x_{n-1})$. One has to consider the first step of integration apart. The error on the first step is bounded by $h f_N(0) \leq h\sigma N$. As for all subsequent steps, n, the error is bounded by $|h^2 x''(a)|$ for some $nh \leq a \leq (n+1)h$.

$$\begin{aligned}h^2|x''(a)| &\leq |h^2 \left[\sigma N \left(1 - \left(\frac{x_n}{C}\right)^\alpha\right) - \theta x_n\right] \left[-\sigma N \alpha \frac{x_n^{\alpha-1}}{C^\alpha} - \theta\right]|, \\ &\leq h^2 \frac{\sigma^{\alpha+1} N^{\alpha+1}}{C^\alpha} \alpha h^{\alpha-1}, \\ &\leq \frac{\sigma^{\alpha+1} N^{\alpha+1}}{C^\alpha} \alpha h^{\alpha+1}.\end{aligned}$$

Therefore, the total error by integrating over a duration T is bounded by

$$e = h\sigma N + \alpha T \frac{\sigma^{\alpha+1} N^{\alpha+1}}{C^\alpha} h^\alpha.$$

B More on the Squirrel Markov chain

The squirrel Markov chain $S_n = (N(T_n), x(T_n))$ is proved uniformly ergodic in Section 4 and $(0,0)$ is an atom. However, its final class (the set of states reachable with positive probability) is not clearly stated. We define $\mathcal{C} = \cup\{A : \exists m | P^m(s, A) > 0\}$ for all $s \in \mathcal{D}$

Lemma 12. *The final reachable class* $\mathcal{C} = \{(N,x)|N \in \{1,\ldots,N_{max}\}\ ,\ 0 < x < \ell_N\} \cup \{(0,0)\} \cup \{(1,0)\} \cup \{(1,0)\}$.

Proof. First note that state $(0,0)$ is reachable from any state $(N,x) \in \{0,\ldots N_{max}\} \times [0,C]$ (see Section 4). Now, from state $(0,0)$, one may reach state (N,x) for any $0 < N \leq N_{max}$ and any x such that $0 < x < \ell_N$ by letting N arrivals come almost simultaneously and let the time run, so that x may grow from 0 to ℓ_N.

This means that $\mathcal{D} \stackrel{\text{def}}{=} \{(N,x)|N \in \{1,\ldots,N_{max}\}, 0 < x < \ell_N\} \subset \mathcal{C}$.

Also, state $(1,0)$ is the first state visited from state $(0,0)$.

Next, we show that during the evolution of the Markov chain starting in $(0,0)$, no state with $x > \ell_N$ may ever be reached. Starting in state (N,x) with $x < \ell_N$, one may only reach (N',x') with $N' = N$, or $N' = N-1$ or $N' = N+1$.

1. If $N' = N$ then $x' < \ell_N$ and the state stays in $\mathcal{D}$.
2. If $N' = N+1$ then $x' = x < \ell_N \leq \ell_{N+1}$ and the state stays in $\mathcal{D}$.
3. If $N' = N-1$ then x' may reach any value between $x(N-1)/N$ and ℓ_{N-1}. The proof is over if one can show that $(N-1)x/N < \ell_{N-1}$.

It remains to show that $(N-1)x/N < \ell_{N-1}$ for any $0 < x < \ell_N$. It is enough to show that $\ell_N N'/N < \ell_{N'}$, with $N' = N-1$.

Let us consider $f_{N'}(\ell_N N'/N)$:

$$\begin{aligned}
f_{N'}\left(\ell_N \frac{N'}{N}\right) &= \sigma N'\left(1 - \left(\frac{N'\ell_N}{NC}\right)^\alpha\right) - \theta\frac{N'}{N}\ell_N, \\
&= \sigma N'\left(\left(\frac{N'}{N}\right)^\alpha\left(1 - \left(\frac{\ell_N}{C}\right)^\alpha\right) + 1 - \left(\frac{N'}{N}\right)^\alpha\right) - \theta\frac{N'}{N}\ell_N, \\
&= \left(1 - \left(\frac{N'}{N}\right)^\alpha\right)\left(\sigma - \frac{\theta\ell_N}{N}\right), \\
&= \left(1 - \left(\frac{N'}{N}\right)^\alpha\right)\sigma\left(\frac{\ell_N}{NC}\right)^\alpha, \\
&\geq 0.
\end{aligned}$$

$f_{N'}(\ell_N N'/N) \geq 0$ means that $\ell_N N'/N < \ell_{N'}$ because $f_{N'}$ is decreasing and $f_{N'}(\ell'_N) = 0$ (see Appendix A.2). □

References

[Alur et al., 2003] Alur, R., Dang, T., Esposito, J., Hur, Y., Ivancic, F., Kumar, V., Lee, I., Mishra, P., Pappas, G. J., and Sokolsky, O. (2003). Hierarchical modeling and analysis of embedded systems. *Proceedings of the IEEE*, 91(1):11–28.

[Asarin et al., 1995] Asarin, E., Maler, O., and Pnueli, A. (1995). Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138:35–65.

[Baccelli and Brémaud, 1994] Baccelli, F. and Brémaud, P. (1994). *Elements of Queueing Theory: Palm-Martingale Calculus and Stochastic Recurrences*. Springer Verlag.

[Breslau et al., 1999] Breslau, L., Cao, P., Fan, L., Phillips, G., and Shenker, S. (1999). Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of IEEE INFOCOM '99*, pages 126–134, New York.

[Clévenot and Nain, 2004] Clévenot, F. and Nain, P. (2004). A simple model for the analysis of the Squirrel peer-to-peer caching system. In *Proceedings of IEEE INFOCOM 2004*, Hong Kong.

[Clévenot-Perronnin, 2005] Clévenot-Perronnin, F. (2005). *Fluid Models for Content Distribution Systems*. PhD thesis, University of Nice-Sophia Antipolis. http://www-sop.inria.fr/dias/Theses/phd-9.php.

[Dai, 1995] Dai, J. (1995). On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Annals of Applied Probability*, 5:49–77.

[David and Alla, 2004] David, R. and Alla, H. (2004). *Discrete, Continuous, and Hybrid Petri Nets*. Springer-Verlag.

[Davis, 1984] Davis, M. H. A. (1984). Piecewise deterministic markov processes: a general class of non-diffusion stochastic models. *Journal Royal Statistical Society (B)*, 46:353–388.

[Foss and Tweedie, 1998] Foss, S. and Tweedie, R. (1998). Perfect simulation and backward coupling. *Stochastic Models*, 14:187–204.

[Gaujal and Perronnin, 2007] Gaujal, B. and Perronnin, F. (2007). Coupling from the past in hybrid models for file sharing peer to peer systems. In LNCS, editor, *Proc HSCC*, Psa, Italy.

[Ghosh et al., 1997] Ghosh, M. K., Arapostathis, A., and Marcus, S. I. (1997). Ergodic control of switching diffusions. *SIAM Journal on Control and Optimization*, 35(6):1952–1988.

[Girard, 2005] Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems : Computation and Control*, volume 3414 of *LNCS*, pages 291–305.

[Iyer et al., 2002] Iyer, S., Rowstron, A., and Druschel, P. (2002). Squirrel: A decentralized, peer-to-peer Web cache. In *Proceedings of of ACM Symposium on Principles of Distributed Computing (PODC 2002)*, pages 213–222, Monterey, California.

[Kelly, 1979] Kelly, F. P. (1979). *Reversibility and Stochastic Networks*. Wiley, Chichester.

[Liberzon, 1973] Liberzon, D. (1973). *Switching in systems and control*. Birkhäuser.

[Press et al., 1992] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C*. Cambridge University Press.

[Propp and Wilson, 1996] Propp, D. and Wilson, J. (1996). Exact sampling with coupled Markov chains and application to statistical mechanics. *Random Structures and Algorithms*, 9(1):223–252.

[Rowstron and Druschel, 2001] Rowstron, A. and Druschel, P. (2001). Pastry: Scalable distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of Int. Conf. on Distributed Systems Platforms (Middleware)*, Heideberger, Germany.

[Tomlin et al., 2003] Tomlin, C., Mitchell, I., Bayen, A., and Oishi, M. (2003). Computational techniques for the verification and control of hybrid systems,. *Proceedings of the IEEE*, 91(7):986–1001.

[Vincent, 2005] Vincent, J.-M. (2005). Perfect simulation of monotone systems for rare event probability estimation. In *Winter Simulation Conference*, Orlando.

[Walker, 1974] Walker, A. (1974). An efficient method for generating random variables with general distributions. *ACM Trans. Math. Software*, pages 253–256.